University of Twente

EEMCS / Electrical Engineering
Control Engineering

Realtime Stereo Vision Processing for a Humanoid

Rob Reilink

MSc report

Supervisors:
prof.dr.ir. S. Stramigioli
ir. G. van Oort
ir. E.C. Dertien
dr.ir. F. van der Heijden

June 2008

Report nr. 019CE2008
Control Engineering
EE-Math-CS
University of Twente
P.O.Box 217
7500 AE Enschede
The Netherlands

Abstract

This report describes the design of a vision system which is used to control a pair of cameras to move in a way like humans move their eyes. An existing algorithm has been used to give a value of 'interestingness' to each region in the image. This algorithm was extended to work in a moving-camera setup. It has been used to control the cameras to attend interesting targets in its environment in a human-looking way. A stereo matching algorithm has been designed to let the two cameras converge such that the attended target is in the center of both camera images, enabling target depth estimation. These two algorithms have been combined resulting in a setup that scans its environment for salient targets and determines their distance.

Preface

After about 9 months, time has come to finish my Masters project. Especially the last month was a busy period, with a demo setup to be built and at the same time a report to be written. In some aspects it was a last moth just like the last moth of other large projects I've been involved with before; the 2005 University of Twente solar car racing team and the 2006 MIT vehicle design summit. Parts that come in late, software that isn't as finished as you thought it was, strange errors caused by something you never thought of. I seem to get used to it.

For sure, I've learnt to be prepared for these issues during those projects. The experience that I gained has helped me to forsee and prevent many potential problems, both on the technical and on the non-technical side of this project. Again, the planning proved crucial.

On the other hand, other things went so much easier than the doom scenario that I had expected. For example, a broken position encoder was repaired by the supplier within a few hours. Also, the integration of two pieces of software on which I and Ludo had been working seperately for months, took less than two days to be combined.

A quite unique aspect of this masters project was the cooperation with two other students, Ludo and Jan. I have really enjoyed this cooperation. It gives you the opportunity to discuss your ideas and your progress (or standstill) at the coffee machine, it allows to design a multi-disciplinary system from different points of view and, most importantly, it now and then gives you the ability to blame someone else.

I've spent quite some time with Ludo programming to get our demo setup to work. I owe him an apology for too often commenting on his lack of nerd-, vi-, programming and soldering skills and for continuously trying to shift work to him because he was ahead of me in writing his papers and his report.

According to the supervisor of a friend of mine, one should not need to explitly thank their supervisors, since it is their job to support you. I will therefore not explicitly thank Stefano for his inspiration and never-ending enthousiasm, Ferdi for his broad knowledge in the field of vision, Gijs for his occasional critical reviews, Edwin for the suggestions on the electronics hardware and Rafaella for her comments on my papers. I've enjoyed our cooperation and look forward to at least the next four years.

Finally, I'd like to thank my family for their ongoing support during my study. Without them, I wouldn't have been where I am now.

Rob
Enschede, June 2008

Contents

1. Introduction

Ever since the beginning of robotics technology, the human being has been the model for robots. Although a humanoid is not the optimal solution for most problems that can be solved by robots, creating a humanoid is the dream of many robotics engineers and scientists. Why would one want to have a robot that looks like a human? Because humans prefer to interact with humans over interacting with a 'machine'. Thus, if interaction with a machine is required, it'd better look and behave like a human. It is really remarkable how easyly humans associate certain motion and behavourial patterns with human characteristics like emotions.

The Control Engineering group at the University of Twente is also active in the development of humanoids. In collaboration with groups at the universities of Delt and Eindhoven, a soccer playing humanoid robot is under development. This project lead to the idea of developing a 'humanoid head': a head that would behave like a human. Already in an early stage it became clear that given the set requirements, this head would not be suitable for the soccer playing robot. It was then decided to focus on developing a stand-alone setup that can be used both for demonstration and for research purposes.

The developed humanoid head consists of a mechanical neck, which has four degrees of freedom, with on top of it a plate with two movable cameras that function as the eyes. A vision processing computer processes the images from the two cameras and sends the location of the most interesting thing in its view to the control computer that controls the motion.

This report describes the vision processing related with this project and is divided into two parts: the target selection and the stereo vision. The target selection deals with extracting 'interesting' regions from the image. In this context, interesting means that some region has a different color, intensity, orientation, etc. than its environment. The stereo vision deals with controlling the angle between the cameras such that they both look at the same target. The target selection and stereo vision have been described in two separate papers which are included on the following pages.

Saliency-based humanoid gaze emulation using a moving camera setup

R. Reilink, S. Stramigioli, F. van der Heijden and G. van Oort

***Abstract*—This paper describes a vision algorithm which is used to control a pair of cameras to move in a way like humans move their eyes. An existing saliency map algorithm is used to give a value of 'interestingness' to each pixel in the input image, which is then used to select the focus target of the cameras. This algorithm was extended to work in a moving-camera setup: because the algorithm relates data from subsequent video frames, the movement of the cameras must be accounted for. To do this, a model of the mapping from points in the environment to CCD pixels is introduced. This model is validated and the behaviour of the complete setup with the adapted saliency algorithm is evaluated.**

I. Introduction

Creating a humanoid requires also mimicking human behaviour. In non-verbal communication, head and eye movements are important factors. Thus, in order to be human-like, a humanoid needs head and eye movements similar to a human being.

A humanoid head-neck system is being developed at the control engineering group at the University of Twente, in collaboration with an industry partner [1],[2]. The purpose of this 'humanoid head' is to research interaction between humanoid robots and humans in a natural way.

To test the mimicking of the human eye movements a mechanical eye setup was built where two cameras can pan individually, but tilt simoultaneously, as shown in figure 1. This setup is similar to that used by Pettersson and Petersson [3]. This setup was chosen so the cameras can converge, to obtain human-like stereo-vision which may be developed in the future. The setup was driven by three digital modelling servos. To improve the dynamic behaviour, cameras were selected that have a separate moving CCD and a stationary processing PCB, connected by a flexfoil. These COTS camera modules are interfaced using Firewire. Currently, only one of the cameras is used as a video input, the other one is just steered to the same orientation.

To determine where the system should look at, an algorithm developed by Itty was used [4]. In the original work it has been used to process static images and computer-generated images [5]. In this paper, we explain how this can be extended to a system in which a moving camera is used as the input source.

This paper is organised as follows: in section II, the saliency algorithm developed by Itty will be described. Then, in section III a model of the system setup will be introduced. Using this model, the effects of the moving cameras on the saliency algorithm and the required algorithm adaptations are discussed in section IV. The model of the system and the adapted algorithm are evaluated using experiments, described in section V,

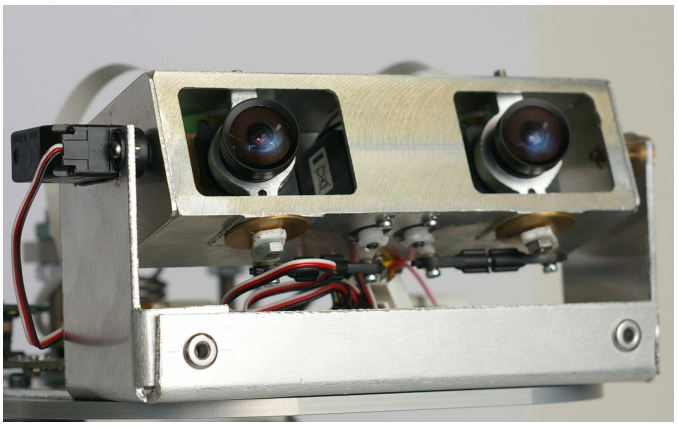


Fig. 1. Camera setup

and finally the results are discussed and suggestions for further research are given in section VI.

II. Background

Human eye movements are steered by two mechanisms: top-down and bottom-up attention[4]. Top-down attention is a deliberate eye movement, that is task-driven (e.g. follow a ball) and requires understanding of the scene. Bottom-up attention, on the other hand, is the unconscious eye movement initiated by visual cues, e.g. movement or bright colors. Bottom-up attention requires no understanding of the scene.

Itty has described a model of human bottom-up attention in various papers [4],[5],[6]. Using this model, he was able to estimate which areas of an image would be considered 'interesting' by humans. The architecture of this algorithm is shown in figure 2. The algorithm works by splitting the input image into different channels (e.g. intensity, color, orientation, motion). These channels are then low-pass filtered on different scales, and the resulting images are subtracted from each other resulting in a set of band-filtered images of each channel. These images are summed across the scales and across the channels, taking into account that images with only a few pop-outs (strong peaks) are more significant than images with numerous pop-outs. The resulting summed image is called the saliency map, which gives a measure of 'interestingness' to each pixel in the input image.

This resulting saliency map $S(x, y)$ is used to determine the 'most interesting' point, the focus of attention (FOA) F. This is done using a winner-take-all (WTA) network which selects the pixel with the highest saliency value as the FOA. Two

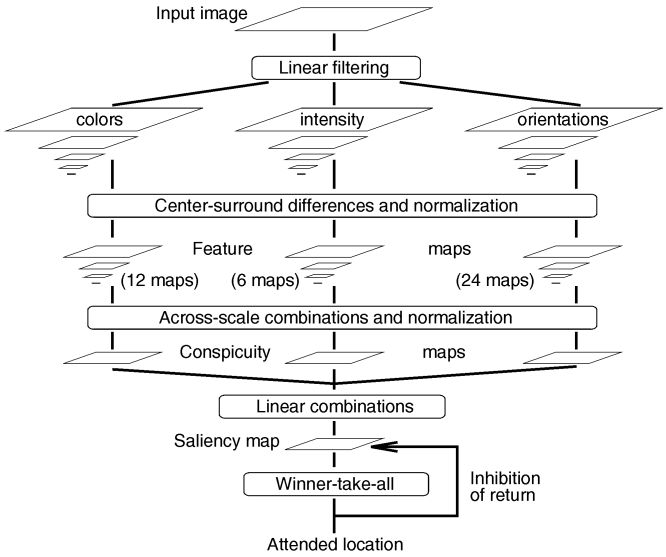


Fig. 2. Architecture of the saliency model [4]

additional mechanisms influence the selection of the FOA: the inhibition of return (IOR) map and the WTA bias.

An IOR map is used to prevent the FOA from staying constant all the time, by giving a negative bias to those regions of the saliency map that were attended recently. This IOR map is a first-order lowpass filter whose input is a Gaussian function $G(x, y)$ positioned at the FOA:

$$G_\sigma(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

$$IOR_{\mathrm{n}}(x,y) = \alpha IOR_{\mathrm{n-1}}(x,y) + \beta G_{\sigma_{\mathrm{IOR}}}(x - F_{\mathrm{n-1,x}}, y - F_{\mathrm{n-1,y}}) \quad (2)$$

The first order difference equation (2) causes the IOR map values to increase around the previous FOA $F_{\mathrm{n-1}}$ while it decays everywhere else ($0 < \alpha < 1$). As a result, the IOR map will have a higher value the longer it was the FOA recently.

The WTA bias $B(x, y)$ is a positive bias given to a region surrounding the previous FOA to create a hysteresis. This prevents jumping between multiple targets with an almost equal saliency. Since not only the previous FOA is biased but also a region around it, a target can also be tracked if it has moved fince the previous frame. The maximum speed at which a target be tracked will be limited by the framerate and the size of the bias.

The saliency map, the IOR map and the WTA bias are summed and fed into the wta network:

$$B_{\mathrm{n}}(x,y) = \gamma G_{\sigma_{\mathrm{B}}}(x - F_{\mathrm{n-1,x}}, y - F_{\mathrm{n-1,y}}) \quad (3)$$

$$F_{\mathrm{n}} = wta(S_{\mathrm{n}}(x,y) - IOR_{\mathrm{n}}(x,y) + B_{\mathrm{n}}(x,y)) \quad (4)$$

Thus, the next FOA target is the most salient location, biased negatively for regions that were recently attended and biased positively to stay at the current location. The constants α,β and γ can be adjusted to influence the dynamic behaviour of the FOA.

When the new FOA target is known, the eyes can be controlled. The setup will behave in one of two modes: tracking or saccade. When tracking, the eyes follow the FOA, which may be moving, using a proportional controller. In a saccade, the eyes move from the previous FOA to the next at their maximum speed. This happens when the distance between the new and the previous FOA is larger than a certain threshold. During a saccade, the camera input is inhibited since it is severely distorted by motion blur.

III. System description and modeling

The image that is captured by the camera is a projection of the environment. The properties of this projection are determined by the camera position and orientation, the lens and the camera itself. In the setup, the camera only rotates around its optical center, it does not translate. The orientation of the camera is assumed to be equal to the setpoints of the servos used to control it; their dynamic behaviour is not modelled. This assumption does not hold during a saccade, when the setpoint changes instantaneously. Therefore, a 300ms settling time is assumed after which the servos will have reached their setpoint.

In order to correct for the effects of the moving camera the transformation from points in the environment to pixels on the camera CCD is modelled. This transformation is a combination of the camera orientation, the perspective transformation and the lens distortion.

A. Coordinate systems

To model the coordinate space transformation, four coordinate systems are used. If we indicate with $\mathcal{E}(3)$ the set of Euclidean points, the ortho-normal world coordinate space map $\Psi^{\mathrm{w}} : \mathcal{E}(3) \to \mathbb{R}^3$ has its origin at the center of rotation of the camera, with the x and y axes parallel to the CCD rows and columns and the z axis pointing out of the camera when the camera is in its neutral position.

The rotated world space map $\Psi^{\mathrm{rw}} : \mathcal{E}(3) \to \mathbb{R}^3$ is Ψ^{w} transformed by the pan and tilt of the camera. The z-axis is the optical axis in the viewing direction of the camera. When the camera is in its neutral position, the Ψ^{w} and Ψ^{rw} coordinate systems coincide.

The corrected image space map $\Psi^{\mathrm{ci}} : \mathcal{E}(3) \to \mathbb{R}^2$ is the ideal perspective projection of Ψ^{rw} on the camera image plane if there was no lens distortion. The lens distortion correction requires the origin of this space to coincide with the optical center of the lens. This is not nescessarily the center of the CCD.

The image space map $\Psi^{\mathrm{i}} : \mathcal{E}(3) \to \mathbb{Z}^2$ is Ψ^{ci} transformed by the lens distortion and is how the world is perceived by the camera CCD. The origin of this space coincides with the origin of Ψ^{ci}.

B. Transformations

The orientation of the camera can be described by its tilt angle θ and pan angle ϕ, which can be used to contruct the rotation matrix R_{tilt} around the x axis and R_{pan} around the y axis. In the used setup, the panning axis is mounted in a frame, which is tilted. The combination of the two rotations results in the transformation T_{r} given as:

$$T_{\mathrm{r}} : \mathbb{R}^3 \to \mathbb{R}^3; p \mapsto R_{\mathrm{pan}} R_{\mathrm{tilt}} p \quad (5)$$

The lens in the camera maps the three-dimensional world onto the two-dimensional image plane. This can be described by the non-linear perspective transformation given by eq. 6. This equation assumes the optical center of the lens to be equal to the center of rotation of the camera. This is not necessarily the case, but since the distance between these centers is in the order of a few milimeters, the resulting camera translation is negligable. The scale factor c is determined by the lens focal distance and the CCD pixel pitch, and can be determined using either lens and CCD specifications or by calibration measurements.

$$T_{\mathrm{p}} : \mathbb{R}^3 \to \mathbb{R}^2; p \mapsto \begin{pmatrix} \frac{cp_x}{p_z} \\ \frac{cp_y}{p_z} \end{pmatrix} \tag{6}$$

The lens distortion caused by the fish-eye lens is modelled as radial distortion [7]:

$$f : \quad \mathbb{R} \to \mathbb{R}; \quad x \mapsto ax^2 + bx \tag{7}$$

$$T_{\mathrm{d}} : \quad \mathbb{R}^2 \to \mathbb{R}^2; \quad p \mapsto f(|p|)\frac{p}{|p|} \tag{8}$$

Here, $|p|$ is the Euclidian norm. A 2nd order polynomial function f was used as the radial correction function. This makes f easily invertible, and calibration measurements showed a 2nd order function is sufficient. The parameters a and b were determined by calibration with a grid pattern. Out of parameters a, b from equation 7 and c from equation 6, one can be chosen arbitrarily.

To invert T_{d}, we set $q = T_{\mathrm{d}}(p)$ and solve p for q;

$$q = T_{\mathrm{d}}(p) = f(|p|)\frac{p}{|p|} \Rightarrow p = |p|\frac{q}{f(|p|)} \tag{9}$$

The norm of q: $|q| = f(|p|)$. Therefore, $|p| = f^{-1}(|q|)$. Substituting these in equation 9 yields:

$$T_{\mathrm{d}}^{-1} : \mathbb{R}^2 \to \mathbb{R}^2; q \mapsto f^{-1}(|q|)\frac{q}{|q|} \tag{10}$$

The distortion model was validated using a 50x50mm spaced grid. Figure 3 shows an image of this grid taken by the camera, together with a grid which was deformed using the lens distortion transformation model. It can be seen that the deformed grid matches the image closely, which shows that the lens distortion matches the model.

The three transformations T_{r}, T_{p} and T_{d} combined describe the mapping from a point in the world p^{w} to the CCD p^{i}:

$$p^{\mathrm{w}} \xrightarrow{T_{\mathrm{r}}} p^{\mathrm{rw}} \xrightarrow{T_{\mathrm{p}}} p^{\mathrm{ci}} \xrightarrow{T_{\mathrm{d}}} p^{\mathrm{i}} \tag{11}$$

IV. Adapting the saliency algorithm to a moving camera

In order to use the saliency algorithm in a system with a moving camera, it must be adapted to take the changing camera orientation into account. This means that all data which is created in one frame and used in another must be transformed according to this change. Also, when a saccade is initiated, the setpoint for the new camera orientation must be calculated using the described model.

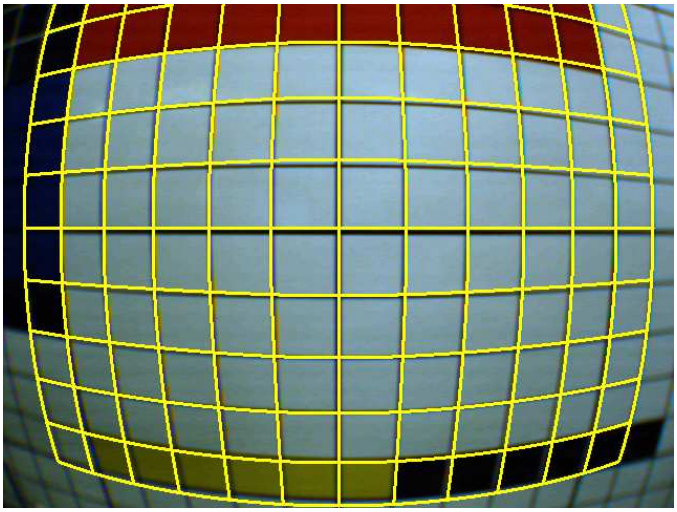


Fig. 3. Barrel distortion correction

A. Feed-forward saccade movement

When a saccade is initiated, the target position is known in image coordinates. A new camera orientation is to be found such that the target position will map to the center of the image $(0,0)^{\mathrm{i}}$. Using the inverse lens distortion transformation, corrected image coordinates of the target are obtained. These cannot be mapped to rotated world coordinates directly because the perspective transformation is not invertible. However, they can be mapped to a plane at $z = d$ which results in

$$p^{\mathrm{rw}} = d \begin{pmatrix} \frac{p_{\mathrm{x}}^{\mathrm{ci}}}{c} \\ \frac{p_{\mathrm{y}}^{\mathrm{ci}}}{c} \\ 1 \end{pmatrix} \tag{12}$$

This leaves the unknown factor d but this will cancel out later since only the orientation of p is of importance. Then, the transformation to world coordinates is straight-forward since the rotation matrices are orthonormal:

$$p^{\mathrm{w}} = R_{\mathrm{tilt}}^{-1} R_{\mathrm{pan}}^{-1} p^{\mathrm{rw}} = R_{\mathrm{tilt}}^{\mathrm{T}} R_{\mathrm{pan}}^{\mathrm{T}} p^{\mathrm{rw}} \tag{13}$$

Now, tilt and pan angles can be calculated such that the coordinates of p after the saccade (denoted by a star) in image space are $p^{\mathrm{i}^*}=(0,0)$, so $p^{\mathrm{rw}^*}=(0,0,z)$. Solving pan and tilt angles ϕ and θ can easily be done geometrically as shown in figure 4. p' is p projected on the world y-z plane. The tilt angle θ is the angle between p' and the world z-axis and pan angle ϕ is the angle between p and p'.

B. IOR map

The inhibition of return (IOR) mechanism causes a certain region to become less 'interesting' when the camera is looking at it. This causes the system to keep scanning its environment instead of staring at a single salient location. The region at which the camera has been looking is defined in the world space, while the processing of the IOR map takes place in the image space. Ideally, every point in space would correspond

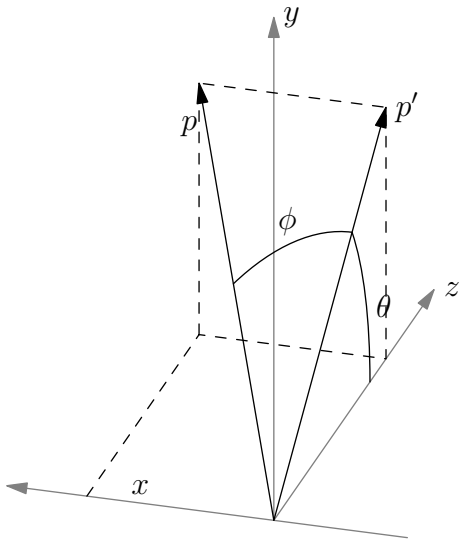


Fig. 4. Obtaining pan and tilt angles from p

to a single pixel on the IOR map, independent of the camera orientation. With a stationary camera, this mapping is

$$p^{\mathrm{w}} \xrightarrow{T_{\mathrm{p}}} p^{\mathrm{ci}} \xrightarrow{T_{\mathrm{d}}} p^{\mathrm{IOR}}; p^{\mathrm{IOR}} = (T_{\mathrm{d}} \circ T_{\mathrm{p}})(p^{\mathrm{w}}), \qquad (14)$$

the same as the mapping from world space to image space when the camera is in its neutral position. To compensate for a moving camera, the transformation from image coordinates to IOR map coordinates, would be:

$$p^{\mathrm{IOR}} = (T_{\mathrm{d}} \circ T_{\mathrm{p}} \circ T_{\mathrm{r}}^{-1} \circ T_{\mathrm{p}}^{-1} \circ T_{\mathrm{d}}^{-1})(p^{\mathrm{i}}). \qquad (15)$$

However, to map every pixel of the image space to the IOR map and back would require an inacceptable amount of processing power. Therefore, for the purpose of the IOR map this transformation is simplified to a shift with respect to the image coordinate space:

$$p^{\mathrm{IOR}'} = p^{\mathrm{i}} + s, \qquad (16)$$

with s chosen such that the center of the image $c = (0,0)^{\mathrm{i}} = (0,0,z)^{\mathrm{rw}}$ maps according to equation 15:

$$c^{\mathrm{IOR}'} = (T_{\mathrm{d}} \circ T_{\mathrm{p}} \circ T_{\mathrm{r}}^{-1}) \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix} = c^{\mathrm{i}} + s = s \qquad (17)$$

with z cancelling out in the perspective transformation T_{p}. Of course, this simplification results in an error in the mapping. A point p will not map to the same pixel in the IOR map when the camera rotates. The IOR map has a low spatial frequency because it is a sum of gaussian functions with a large σ and therefore has a limited gradient. Therefore, the error

$$e_{\mathrm{IOR}} = |IOR(x,y) - IOR(x + \Delta x, y + \Delta y)| \qquad (18)$$

is also limited.

C. WTA bias

When determining the maximum salient location in the WTA stage, a bias is applied to the position of the estimated FOA target to create a hysteresis. Like the IOR map, this estimated position is defined in the world space, and a transformation to image coordinates is required. Because only a single point needs to be transformed, the actual transformation and its inverse can be used; the simplification as done with the IOR map is not necessary. However, the simplification might be acceptable since the WTA bias is also a gaussian function.

The FOA of the previous frame is known in image coordinates, F_{i}. This is transformed to world coordinates using the pan and tilt angle at the time of that frame (T_{r}^{-1}), and transformed back to image coordinates of the current frame $F_{\mathrm{i}*}$ using the current pan and tilt angles ($T_{\mathrm{r}*}$) :

$$F^{\mathrm{i}*} = (T_{\mathrm{d}} \circ T_{\mathrm{p}} \circ T_{\mathrm{r}*} \circ T_{\mathrm{r}}^{-1} \circ T_{\mathrm{p}}^{-1} \circ T_{\mathrm{d}}^{-1})(F^{\mathrm{i}}) \qquad (19)$$

D. Motion and flicker channels

The saliency map algorithm described in [5] also incorporates motion and flicker channels which react to changes in the image. These channels require image data from previous frames. This means these channels must be adapted to take the camera orientation into account. Since the image data may have a high spatial frequecy, an accurate transformation might be required, which could result in a high computational load. Since the motion and flicker channels were not used in this setup, the required adaptations were not investigated.

V. Evaluation

The algorithm was evaluated by two experiments, validating the transformation model and testing the saliency model on the moving camera setup. Since a static stimulus was used to test the saliency model, the tracking could not be tested. Simple tests showed that the setup could track a moving salient object, but a more elaborate experiment would be required to quantify the performance of the system, for example in terms of the maximum attainable tracking speed. A projector could be used to project a pre-recorded stimulus on a white screen at which the setup is looking to obtain repeatable results.

A. Transformation model

The transformation model was validated using the feed-forward saccade algorithm. Manually, a fixed point in the environment was picked in the camera image and the required camera movement to get this point in the center of the image was calculated. After this movement was executed, the same point was picked again and the distance between the point and the center of the image was measured. This was repeated several times.

Figure 5 shows the results of the transformation model evaluation. This graph shows the target error as a function of the saccade distance. The target error is the distance between the center of the image and the location of the selected target after the saccade. The saccade distance is measured as the sum of the absolute tilt and pan angle change required for the saccade. The target error was determined with an accuracy in

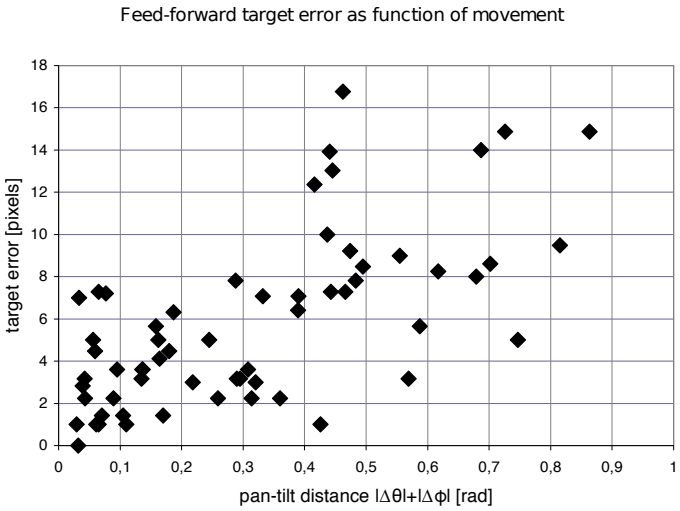


Fig. 5. Error of the feed-forward saccades

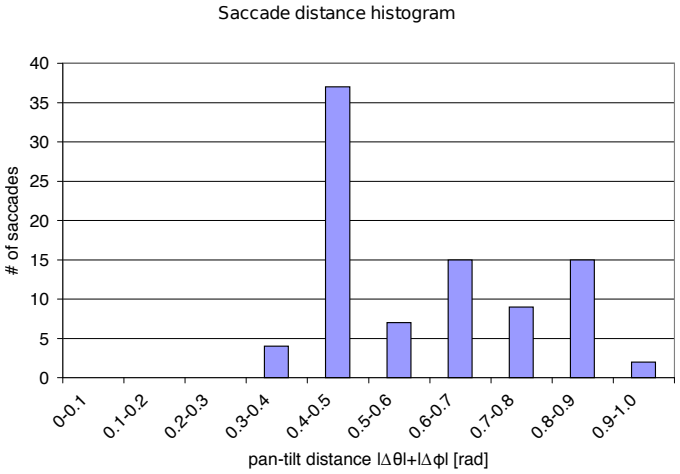


Fig. 6. Histogram of the saccade distance with the saliency algorithm controlling the camera

the order of 3 pixels, as the reference points were manually picked. Clearly, the error depends on the saccade distance. Saccade distances that were recorded with the setup looking into our control engineering lab are shown in a histogram in figure 6. The results show that in this experiment, most saccades had a distance of 0.4-0.5 radians, thus errors of over 15 pixels may be expected.

B. Saliency algorithm

The saliency algorithm is more difficult to evaluate. Because 'human-like' is not a criterion which can be measured easily and objectively, the system was evaluated using an abstract stimulus shown in figure 7. This stimulus was drawn on an A0-sized poster, which was setup such that the system could not see past the borders given its limited mechanical range. A comparison was made between two setups: using a fixed camera and using a moving camera. For these situations, the trace of the FOA was recorded and it was eveluated how often the FOA visited each spot of the image. Also, it was measured how many frames where required for the system to have attended all seven dots in the stimulus.

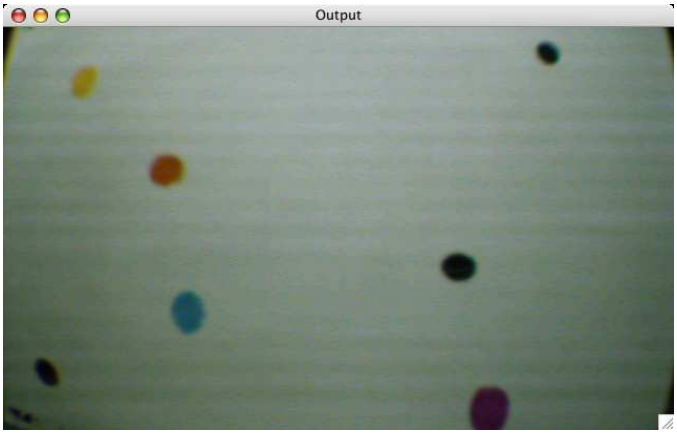


Fig. 7. The stimulus used to test the saliency algorithm

The results are shown in figure 8 and figure 9. These figures show the total time each point was visited: the darker the figure, the more time it was the FOA. To allow a good comparison between the two experiments, care should be taken to keep the boundary conditions and the lighting the same. This was not the case in the experiment, so a more accurate experiment could improve the comparison. This was not possible however due to time constraints.

In the two figures, is clearly visible that in the dynamic situation, the FOA visits areas other than the dots more often than in the static situation. This is partly caused by lighting conditions (shadows), but also by the limited view: when only one dot is visible and the IOR causes the FOA to shift away from this dot, there may be no other dots in the view, causing the FOA to shift to other locations.

In the figures, there is also a trace of the FOA from the start of the test until six of the seven dots where found. The rest of the trace was left out because otherwise the figure would become too cluttered. It is clearly visible that in the static situation, the points are visited sequentially, and the FOA shifts to the correct position right-away. In the dynamic situation, the FOA sometimes shifts from one point to the other rapidly. This is because when the saccade is executed, other areas become visible, which may be even more salient than the original saccade target. Because not all dots are in view simultaneously, they are not visited sequentially. This causes the system to take more time before all dots have been found: 260 frames for the moving camera versus 135 frames for the static situation.

VI. Discussion

The saliency algorithm provides an extensible framework that may be used to perform numerous tasks, depending on the input channels. With apropriate filters, it could be used to find faces or certain objects. Also, the influence of the existing filters could be adjusted to create a form of top-down attention, as described in [6]. The filters could be made time-dependent, to adapt the system to a certain task while it is operating.

Other types of sensors could also be connected to the system, for example as proposed by R. Brooks [8]. Especially

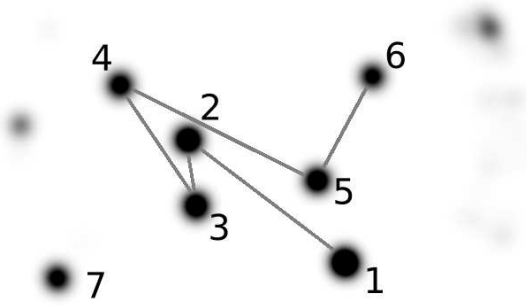


Fig. 8. FOA trace using a fixed camera

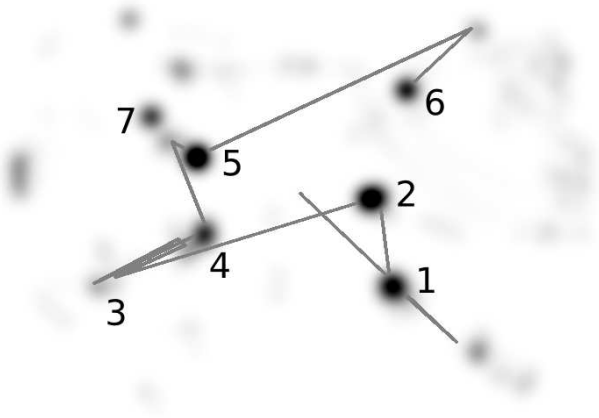


Fig. 9. FOA trace using a moving camera

auditory inputs are interesting, because these are important cues for attention.

A transformation model has been designed which was used to modify the existing saliency algorithm to work in a moving camera setup. The evaluation of the transformation model showed that errors of over 15 pixels can be expected when a saccade is done. The WTA bias spot should be large enough to make sure that the cameras will track to the most salient location after the saccade.

To implement the saliency algorithm on a different setup, e.g. the 3TU humanoid [9] or the humanoid head [1],[2] which are currently under development at our control engineering group, the transformation models of these setups could be used to perform the transformations required in the modified saliency algorithm.

When the algorithm is used in a setup where the position and orientation of the eyes with respect to the world is not fixed, inertial sensors may be used to estimate the transformation matrices.

The saliency algorithm applied to the moving-eye setup has been tested by comparing the FOA trace in a static and in a moving camera setup. Lighting disturbances made these comparisons more difficult. Possibly, the implementation and testing traject could be facilitated by first performing a test in a simulated environment. A software program could be used to generate the camera images from a virtual 3D world using models of the mechanical setup and the camera. Using co-simulation [10], the saliency algorithm, the dynamics of the mechanical system and the simulated environment could be tested as a complete system.

The saliency experiment shows there is a significant difference between a static and a moving camera setup. This is mostly caused by the fact that not all points are visible all the time, and thus only salient points within the current view can be selected as the new FOA: the new FOA will be the most salient point in the current camera view. This also means that a salient point within a large non-salient region may never be seen at all. A bias could be added to force the system to scan its entire mechanical range to ensure every salient point can be attended.

In a more elaborate experiment, the saliency algorithm could also be compared to a human using an eye tracker. However, care should be taken to select the stimulus such that primarily bottom-up attention is stimulated. Since humans have both bottom-up and top-down attention, but the algorithm only implements bottom-up attention, a stimulus that stimulates top-down attention, for example written text, would make the comparison between the human and the algorithm very difficult.

APPENDIX

TABLE I
LIST OF SYMBOLS

$S_{\mathrm{n}}(x,y)$	Saliency map of frame n
$IOR_{\mathrm{n}}(x,y)$	Inhibition of return map from frame n
$B_{\mathrm{n}}(x,y)$	WTA bias from frame n
$G_{\sigma}(x,y)$	Unity-amplitude 2-D Gaussian function with standard deviation σ
F_{n}	Focus of attention location on frame n
T_{r}	Camera rotation transformation
T_{p}	Perspective transformation
T_{d}	Lens distortion transformation
p^{w}	Point in world coordinates
p^{rw}	Point in rotated world coordinates
p^{ci}	Point in corrected image coordinates
p^{i}	Point in image coordinates

REFERENCES

[1] L. Visser, "Motion control of a humanoid head," Masters thesis (unpublished), University of Twente, 2008.

[2] J. Bennik, "Mechatronic design of a humanoid head and neck," Masters thesis (unpublished), University of Twente, 2008.

[3] N. Pettersson and L. Petersson, "Online stereo calibration using fpgas," *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 55–60, 6-8 June 2005.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intenlligence*, vol. 20, no. 11, pp. 1254–1259, November 1998.

[5] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," in *Proc. SPIE 48th Annual International Symposium on Optical Science and Technology*, B. Bosacchi, D. B. Fogel, and J. C. Bezdek, Eds., vol. 5200. Bellingham, WA: SPIE Press, Aug 2003, pp. 64–78.

[6] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2049–2056, 2006.
[7] S. Shah and J. Aggarwal, "A simple calibration procedure for fish-eye (high distortion) lens camera," *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 3422–3427 vol.4, 8-13 May 1994.
[8] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, vol. 2, no. 1, pp. 14–23, Mar 1986.
[9] "Dutch robotics," 2008. [Online]. Available: http://www.dutchrobotics.net
[10] A. Damstra, "Virtual prototyping through co-simulation in hardware/software and mechatronics co-design," Masters thesis (unpublished), University of Twente, 2008.

Focus of attention distance estimation in a saliency-controlled stereo moving camera setup

R. Reilink, S. Stramigioli, F. van der Heijden and G. van Oort

***Abstract*—This paper describes the coupling of a stereo matching algorithm to a saliency algorithm in a stereo moving camera vision system. An existing saliency algorithm, which assigns a value of 'interestingness' to each pixel in the input image, is used to aim the cameras at a target. This is extended with a stereo matching algorithm which is designed to let the cameras converge such that this target is in the center of both camera images. The convergence angle is used to determine the target distance. These two algorithms are combined resulting in a proof-of-principle setup that scans its environment for salient targets and determines their distance.**

I. Introduction

Creating a humanoid requires mimicking human behaviour. Because head and eye movements are important factors in non-verbal communication, such movements are required in a humanoid to mime human behaviour.

A humanoid head-neck system is being developed at the control engineering group at the University of Twente, in collaboration with an industry partner [1],[2]. The purpose of the 'humanoid head' project to interact with humans in a natural way.

To test the mimicking of the human eye movements a stereo vision system was built in which two cameras can pan individually, but tilt simoultaneously, as shown in figure 1. This setup is similar to that used by Pettersson and Petersson [3]. Using this setup, the cameras can converge to look at the same object. The angle between the cameras while they are aimed at the same object can be used to estimate the distance of this object.

In a previous paper, we have described how this setup was used to obtain human-looking eye movement using one camera [4]. An algorithm developed by Itty was used [5], which was modified to work with a system in which a moving camera is used as the input source. In this paper, we show how this can be extended to a system where both cameras are used.

This paper is organised as follows: in Section II some background information on the used saliency algorithm, the stereo correspondence problem and epipolar geometry are given. Then in Section III the design of the system is discussed. Finally, the system is evaluated in section IV and in Section V the results are discussed and suggestions for further research are given.

II. Background

A. Saliency map

To control the movement of the cameras, and determine where the setup should look at, a saliency map was used. This map assigns to each image pixel a value of 'interestingness', based on the spatial frequency content. The input image is split into several channels, e.g. intensity, color and orientation. These are low-pass filtered to different scales, and the filtered images are subtracted from each other resulting in a set of band-filtered images. These are then combined resulting in the saliency map [5].

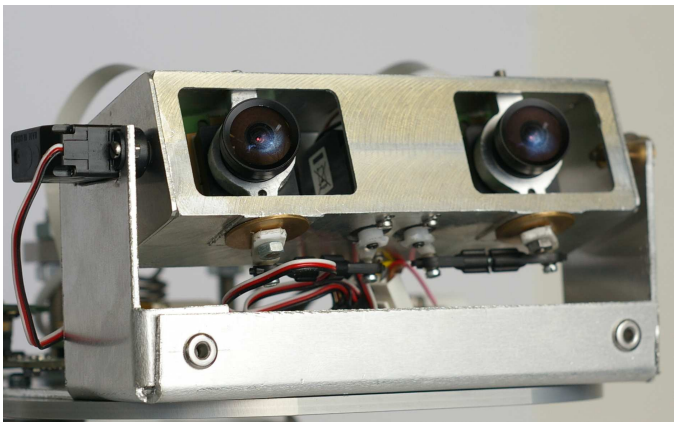


Fig. 1: Camera setup

The saliency map is used to select the next focus of attention (FOA). The cameras move to this point using either a saccade or tracking. A saccade is a movement from one FOA to the other at the maximum speed using a feed-forward set point, whereas with tracking the cameras follow the FOA using a proportional feed-back controller.

B. Correspondence problem

When a FOA target is found, its distance can be estimated. Extracting depth information from two or more cameras requires solving the 'correspondence problem'. The correspondence problem "consists in establishing which point in one image corresponds to which point in another, in the sense of being the image of the same point in space" [6]. Although humans seem to solve this problem effortlessly, the solution is not trivial. Humans use many different clues, like the image context and prior information on the scene. These clues are not usable by a computer which has no understanding of the images.

A common approach in computer vision to solving the correspondence problem is to perform matching over a window: Given a window $W_{\mathrm{ref}}(p_{\mathrm{ref}})$ surrounding a point p_{ref} in the reference image, find a matching point p_{match} with surrounding window $W_{\mathrm{match}}(p_{\mathrm{match}})$ which minimizes cost function

$f(W_{\mathrm{ref}}, W_{\mathrm{match}})$. Common cost functions are the sum of squared differences (SSD) and sum of absolute differences (SAD) between the pixel intensities[7].

C. Epipolar geometry

In the correspondence problem, the geometry constrains the possible matches between two points in the two images. If a point p produces two images p_1, p_2 on two cameras, the positions of p_1 and p_2 are related by the epipolar constraint [6]. For a given camera position, this constraint maps to each point in an image an epipolar line on the other image on which all possible matches lie. Thus, if the epipolar geometry is known, the search for a match can be limited to a given line instead of a search over the entire image. This limits the number of possible false matches, and significantly decreases the required computational power.

The epipolar constraint is described by the essential matrix E, which depends on the camera setup. To calculate a depth map from a set of stereoscopic images, E must be known. The required calibration can be done beforehand if the camera setup is static, but for moving cameras this is not feasible. E can be estimated by first extracting a set of features from both images and matching these [3]. However, this is a process which is computationally quite intensive.

III. Design

The stereo vision algorithm was designed as an extension to the existing saliency-controlled system. This design method of creating a complex system by gradually adding behavioural modules was proposed by Brooks [8]. It provides a way to incrementally build and test a complex robot system. The saliency algorithm controls both cameras, while the stereo algorithm adds a bias to the panning of camera 2 to control the convergence angle.

The primary goal of the designed system is to interact with humans by emulating the human eye movements, including converging the eyes towards the focus of attention. Humans can estimate the gaze direction with an accuracy of about 4° [9]. Thus, the setup should be able to focus at the target with an accuracy of over 4°.

Becuase the avaliable time was limited, only a proof-of-principle was built. Therefore, the focus of the design has been on the coupling of the saliency and the stereo algorithms. The actual stereo matching algorithm and the robustness of the system have not been examined thoroughly.

A. Epipolar geometry

As mentioned before, the epipolar constraint can be used to limit the possible location of the FOA in the second camera to a line. However, the essential matrix E which describes the epipolar geometry is not known on beforehand because the cameras move. Estimating E from the images from both cameras is possible, but is computationally expensive. However, for our system it is not necessary to estimate E since we are only interested in estimating the depth of the FOA. Since camera 1 is controlled by the saliency algorithm, the FOA is the center of the image of camera 1.

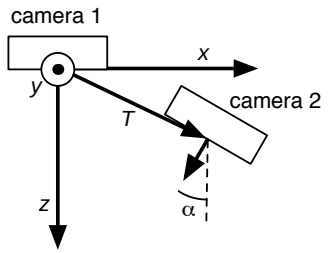


Fig. 2: Position of camera 2 with respect to camera 1 is composed of translation T in the xz-plane and rotation α around the y-axis

Since the two cameras share the same tilt axis, the position of camera 2 in frame 1 can be expressed as a translation T in the xz-plane and a rotation α around the y-axis (figure 2). If the rotation is described by rotation matrix R and $\hat{T}$ denotes the matrix form of the vector product operation, the resulting essential matrix $E \doteq \hat{T}R$ [6] is:

$$E = \begin{pmatrix} 0 & -T_{\mathrm{z}} & 0 \\ T_{\mathrm{z}}\cos\alpha - T_{\mathrm{x}}\sin\alpha & 0 & -T_{\mathrm{z}}\sin\alpha - T_{\mathrm{x}}\cos\alpha \\ 0 & T_{\mathrm{x}} & 0 \end{pmatrix} \tag{1}$$

If $x_1', x_2' \in \mathbb{R}^3$ are the homogeneous pixel coordinates of respectively the FOA projected on camera 1 and the matching point to be found on camera 2, x_1' and x_2' must satisfy: [6]

$$x_2'^{T} K_2^{-\mathrm{T}} E K_1^{-1} x_1' = 0, \tag{2}$$

with K_1 and K_2 denoting the intrinsic parameter matrix of camera 1 and 2, respectively. The intrinsic parameters can be estimated using camera calibration. Then, the calibrated camera coordinates $x_{1,2}$ can be calculated using

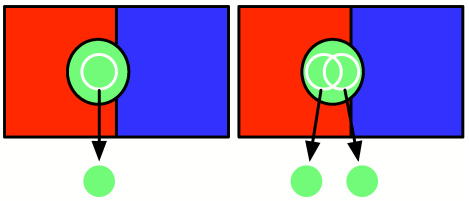
$$x_{\mathrm{i}} = K_{\mathrm{i}}^{-1} x_{\mathrm{i}}', \quad i = 1, 2 \tag{3}$$

Because the FOA is in the optical center of camera 1, x_1=$(0, 0, x_{1,\mathrm{z}})^{\mathrm{T}}$. This yields $x_{2,\mathrm{y}} = 0$, thus the matching point of the optical center of camera 1 will be $(x_{2,\mathrm{x}}, 0, x_{2,\mathrm{z}})^{\mathrm{T}}$ on the x-axis of the calibrated image of camera 2.

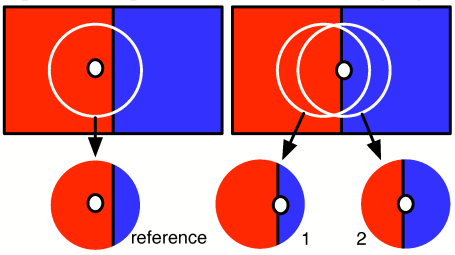
B. Matching function

As mentioned before, finding the best point p_{match} with surrounding window W_{match} to match reference W_{ref} surrounding a point p_{ref} is done by minimizing cost function $f(W_{\mathrm{ref}}, W_{\mathrm{match}})$. The size of the window is an important factor: if the window is smaller than the object being looked at, the location of the best match may be ill-defined. On the other hand, if the window is too large, the best match may be a match of the background instead of the foreground. While the setup is in use, a typical scene will consist of humans standing in front of it, with a distant background. In this case, there will be a clear distinction between the foreground and the background.

This is illustrated in figure 3. This figure shows a left and a right camera view for two situations: a window which is too small (3a) and a window which is too large (3b). In the images, the dark rectangles represent the background and the

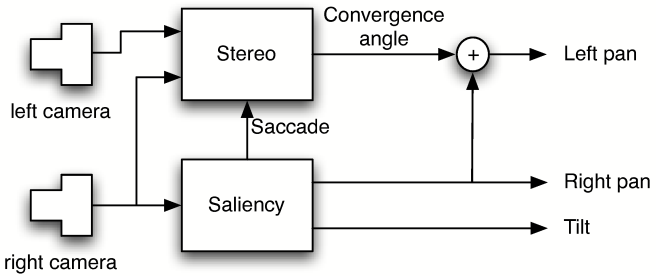


(a) Window too small: ill-defined match. Both possible images on the right match the left window equally well.

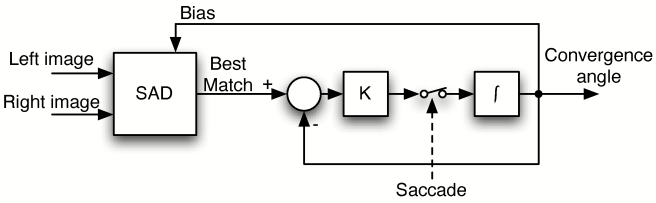


(b) Window too large: background is matched because a background mismatch causes a high cost.

Fig. 3: A window size appropriate to the object size is crucial for correct stereo matching.



(a) Coupling of the stereo matcher and the saliency algorithm



(b) The stereo algorithm consists of a SAD matching function and a proportional controller which is inhibited during saccade.

Fig. 4: Implementation of the stereo matching.

white circles represent the window. The task of the stereo matching algorithm is to select the window in the right image to match the left image.

In figure 3a, the green circle is the target. However, both white circles on the right match the circle on the left equally well and thus the best match will be determined by measurement noise, which is of course undesirable.

In figure 3b on the other hand, the window is too large. Here, the small white dot is the target. Match 2 is the desired match where the target is in the center just as in the reference image. However, the cost for the mismatch of the background between 2 and the reference will be larger than the cost of the mismatch between the target between 1 and the reference, and thus match 1 will have the lowest cost and will be selected as the best match. This results in the cameras looking at the background instead of the object.

Thus, the best window size depends on the scale of the target. The scale of the target is not known beforehand, but the saliency map could help in selecting the appropriate window size by examining the size of the salient region that surrounds the FOA. However, due to time constraints, this has not been implemented yet; a fixed window size apropriate for the set of test object has been chosen manually.

C. Coupling with the saliency algorithm

Since both the saliency and the stereo algorithm need to control the camera orientation, their outputs have to be combined. In our setup, the saliency algorithm determines the FOA target and controls the right camera to look at it, while the stereo algorithm controls the convergence angle, the angle between the optical axes of the two cameras, to follow the FOA target. This is shown in figure 4a. The left camera pan angle is the sum of the right camera pan angle and the convergence angle. This way, if the saliency algorithm causes the FOA to shift, both cameras rotate over the same angle, keeping the convergence angle constant. This means that as long as the FOA distance is constant, it will stay in the center of both cameras without requiring a correction by the stereo algorithm.

To control the convergence angle, a proportional controller is used as shown in figure 4b. This controller uses the best SAD match for the FOA as an input to steer the center of the camera towards the FOA. The proportional constant K cannot be set too high, since there is a delay of 2-3 frames between the frame capture and the result of the processing which would cause instability.

When the best match between the left and right camera is known, this is used to control the cameras such that they both look at the same target. However, when multiple almost-equal matches exist, this might cause the camera to jump between them. To prevent this undesired behaviour, a bias is added to create hysteresis. The SAD matching function is biased towards the current convergence angle. Although this might cause the algorithm to keep focussed at the wrong match, we consider this behaviour better than the cameras moving back and forth between two matches.

When a saccade is taking place, the image of both cameras becomes severely distorted by motion blur. This prohibits proper stereo matching. Also, after a saccade the new target may be at a different distance than the previous target. Thus, it may be useful to have the stereo algorithm to react to a saccade. In the current setup, the stereo matching output is inhibited during the saccade as depicted by the switch in figure 4b. Thus, the convergence angle remains constant during the saccade. After the saccade, the cameras start to converge towards the FOA target. A more sophisticated approach could be to start with a feed-forward controlled camera movement after the saccade: the best stereo match is used to aim the camera at it rightaway. However, during this movement the image will again be blurred due to the motion, preventing stereo matching. Moreover, after the camera has reached its

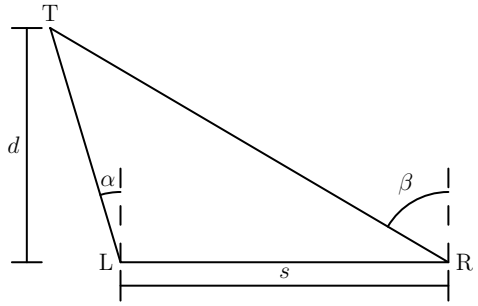


Fig. 5: Estimating the target distance d using triangulation

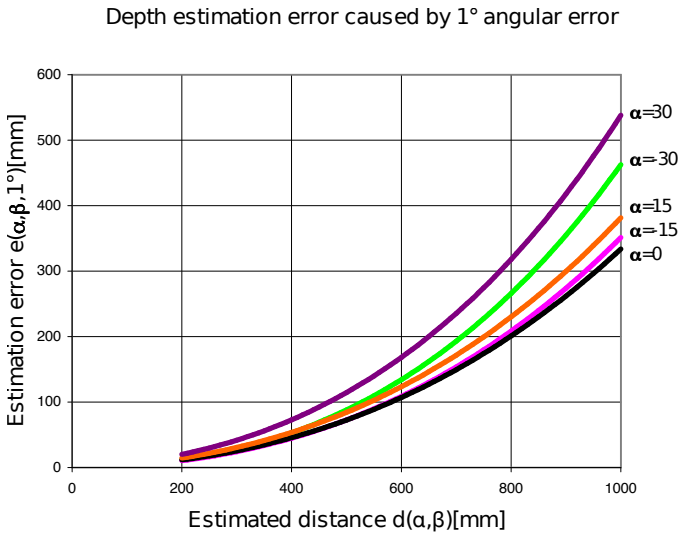


Fig. 6: Estimating error caused by a 1° angular error

setpoint, the target may have moved already. Thus, this is not a straight-forward approach.

D. Depth estimation from convergence angle

When the angles of the two cameras looking at the same object are known, the depth of the object can be estimated using triangulation as shown in figure 5. In this figure, the left camera L and the right camera R are aimed at target T. Distance d between T and the camera baseline can be expressed as a function of angles α and β and camera-spacing s:

$$d(\alpha,\beta) = \frac{s}{\tan(\beta) - \tan(\alpha)} \tag{4}$$

The accuracy with which the target distance can be estimated depends on the accuracy of α and β. Since the anglular positions of the pan actuators are used to determine α and β, an angular alignment error between the actuator and the camera will result in an error in estimating distance d. The resulting distance error e_d can be calculated for an angular error of δ in both positive and negative direction:

$$e_{d+}(\alpha,\beta,\delta) := |d(\alpha,\beta) - d(\alpha,\beta+\delta)| \tag{5}$$
$$e_{d-}(\alpha,\beta,\delta) := |d(\alpha,\beta) - d(\alpha,\beta-\delta)| \tag{6}$$
$$e(\alpha,\beta,\delta) := \max(e_{d+}(\alpha,\beta,\delta), e_{d-}(\alpha,\beta,\delta)) \tag{7}$$

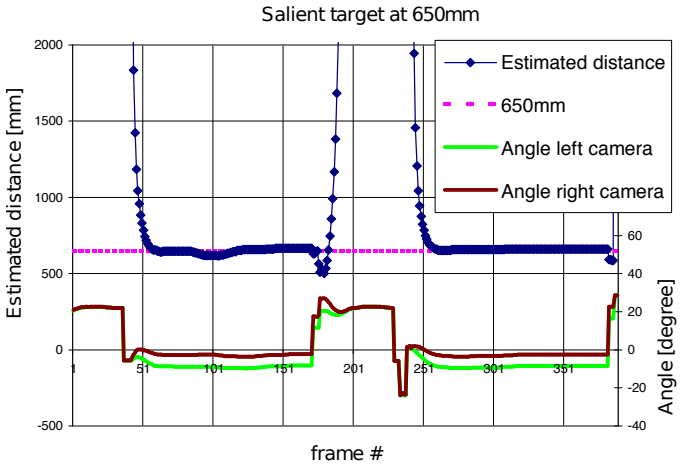


Fig. 7: Camera movements and distance estimation with salient object at 650mm

Figure 6 shows the maximum distance error e in relation to target distance d for different values of α. This figure can be used to get an indication of the attainable accuracy. For this figure, $\delta = 1^\circ$ and $s = 70\text{mm}$ (the actual camera distance in the setup). It is clearly visible that the triangulation works best at small distances. For large distances, a small angular error results in a large error in distance. For example, for $\alpha = 0$ and $s = 70\text{mm}$, $\beta = 1^\circ$ equals 4m while $\beta = 0$ equals an infinite distance. Since the angular accuracy in the used setup may well be worse than 1°, distances of over 4m cannot be measured correctly.

IV. Evaluation

For the evaluation of the setup, a preliminary experiment was done to test the stereo algorithm in combination with the saliency algorithm. More elaborate experiments, for example to assess the robustness of the stereo algorithm, were not possible within the limited timeframe.

For the experiment, the camera setup was put in an environment with a single salient object in the foreground and some less salient objects in the background. Using this setup, the saliency algorithm would cause the system to scan these targets and thereby change its attention from the background objects to the foreground object and vice versa. This was done twice with the foreground object on a different distance from the camera setup: 350 and 650mm. For the experiment with the target at 650mm, figure 7 shows the angles of the left and right camera on the bottom, and the estimated target distance at the top, with the actual distance marked as the dashed line. It shows at frame # 40 how the cameras perform a saccade from a background object to the foreground object, and then start to converge. At frame # 170 the cameras move to a background object again and move to a parallel position. At frame # 230 two saccades take place in succession. This may occur when new targets come into the field of view because the camera has moved [4]. Finally, at frame # 240 the foreground object has the focus again and the cameras converge again.

The results show that the distance of the object is estimated within a 35mm accuracy. This is well within the 125mm error

expected from figure 6. The figure also shows that it takes a settling time of about 20 frames before the estimated distance is correct. For the experiment with the target at 350mm, the distance error was within 30mm. Again, this is within the error expected from figure 6.

V. Discussion

The saliency algorithm has been combined with a stereo matching algorithm to create a stereo camera setup where the cameras move and converge. The convergence angle can be used to estimate the distance of the FOA target. However, the accuracy is limited for distances over 1m. Also, the robustness of the system has not been assessed. This could be an interesting subject of further research.

A relatively simple matching algorithm was used to obtain depth estimation, as the main target was not to build a complete stereo map, but to control the camera movements in a human-like manner. Of course, the field of stereovision is very broad and a more sophisticated depth estimation algorithm could provide a humanoid with very valuable information. This should be considered to be implemented in combination with the saliency algorithm.

The current implementation uses the two cameras in a very different way: one is used for the saliency algorithm and the other one for the stereo matching. This is most likely not how the visual information is processed in the human brain, which might result in a behaviour that is not human-like. It might be possible to join the two camera images and use them both for saliency processing, but this would require much more computational power.

When information of both cameras is processed by a saliency algorithm, it might also be possible to use the saliency map to perform the stereo matching. Instead of the two camera images, two saliency maps derived from these images might be matched. Alternatively, the saliency algorithm may be used to determine which of the input channels (intensity, color, orientation etc.) provides the most distinguishing features, and then that channel may be used for matching.

References

[1] L. Visser, "Motion control of a humanoid head," Masters thesis (unpublished), University of Twente, 2008.

[2] J. Bennik, "Mechatronic design of a humanoid head and neck," Masters thesis (unpublished), University of Twente, 2008.

[3] N. Pettersson and L. Petersson, "Online stereo calibration using fpgas," *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 55–60, 6-8 June 2005.

[4] R. Reilink, S. Stramigioli, F. van der Heijden, and G. van Oort, "Saliency-based humanoid gaze emulation using a moving-camera setup," *(unpublished)*, 2008.

[5] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intenlligence*, vol. 20, no. 11, pp. 1254–1259, November 1998.

[6] Y. Ma, S. Soatto, J. Košecká, and S. Shankar Sastry, *An Invitation to 3-D Vision*. Springer, 2006.

[7] L. Di Stefano, M. Marchionni, and S. Mattoccia, "A pc-based real-time stereo vision system," *Machine Graphics & Vision*, vol. 13, no. 3, pp. 197–220, 2004.

[8] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, vol. 2, no. 1, pp. 14–23, Mar 1986.

[9] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," in *Proc. SPIE 48th Annual International Symposium on Optical Science and Technology*, B. Bosacchi, D. B. Fogel, and J. C. Bezdek, Eds., vol. 5200. Bellingham, WA: SPIE Press, Aug 2003, pp. 64–78.